

Open Source im DLR

Andreas Schreiber

Chaosdorf Düsseldorf, 28.10.2015



Wissen für Morgen





„Open Source im DLR“ von [Andreas Schreiber \(DLR\)](#) ist lizenziert unter einer [Creative Commons Namensnennung 4.0 International Lizenz](#).



Alle Bilder: [DLR](#) (CC-BY 3.0), soweit nicht explizit anders gekennzeichnet.



Überblick

- Software im DLR
- Software Engineering
- Open-Source-Strategie
- Migration nach Python
- Beispiele für Open-Source-Software



Forschung im DLR

Forschungsgebiete

- Raumfahrt
- Luftfahrt
- Energie
- Verkehr
- Sicherheit



Softwareforschung- und Entwicklung

- Einrichtung Simulations- und Softwaretechnik
- Softwareentwicklung in allen 33 DLR-Instituten



Einrichtung Simulations- und Softwaretechnik

- Zentrale Einrichtung mit Softwarekompetenz
- Standorte
 - Berlin-Zentrum
 - Braunschweig
 - Köln-Porz
- Ca. 50 Mitarbeiter
- <http://www.DLR.de/sc>



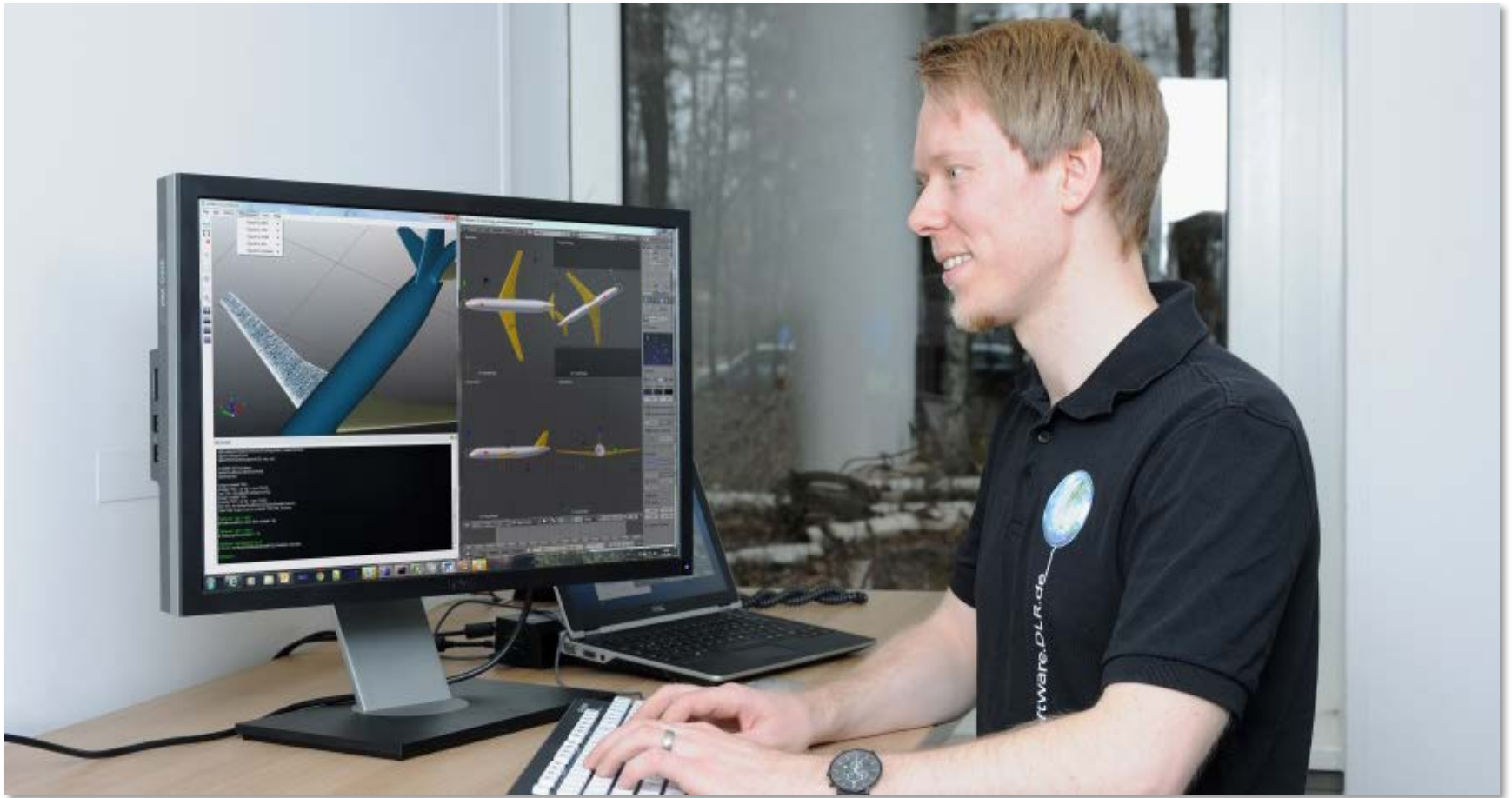
Software-Forschung

Erforschen und Entwickeln neuer Softwaretechnologien



Software-Entwicklung

Individualsoftware für die Forschung im DLR



Software-Qualität

Verbessern der Qualität der Software im DLR



DLR Simulations- und Softwaretechnik

Aktuelle Forschungsgebiete

Big Data Usability Quantencomputing
Multidisziplinäre Simulation

Modellbasiertes Systems Engineering Mobile Systeme

Sicherheit High Performance Computing
Software Engineering Wissensmanagement

Verteilte Systeme Interaktive Visualisierung
Virtual und Augmented Reality Eingebettete Systeme

Datenmanagement



Software im DLR

Umfang

Einige Zahlen...

- Mehr als 1200 Mitarbeiter entwickeln Software
- Mehr als 100 Mio EUR Personalkosten pro Jahr für Softwareentwicklung
- DLR ist eines der größten „Softwarehäuser“ in Deutschland



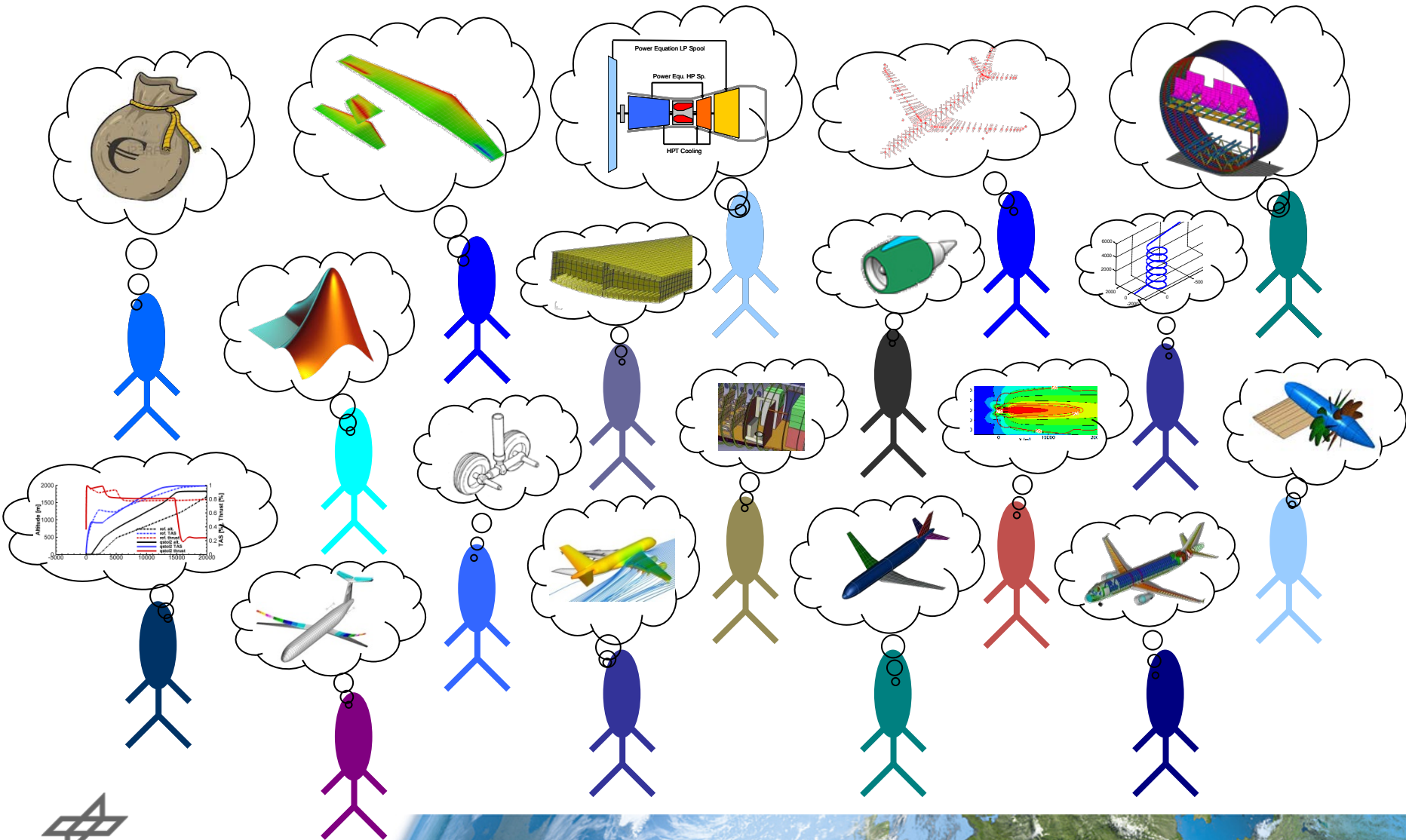
Software im DLR

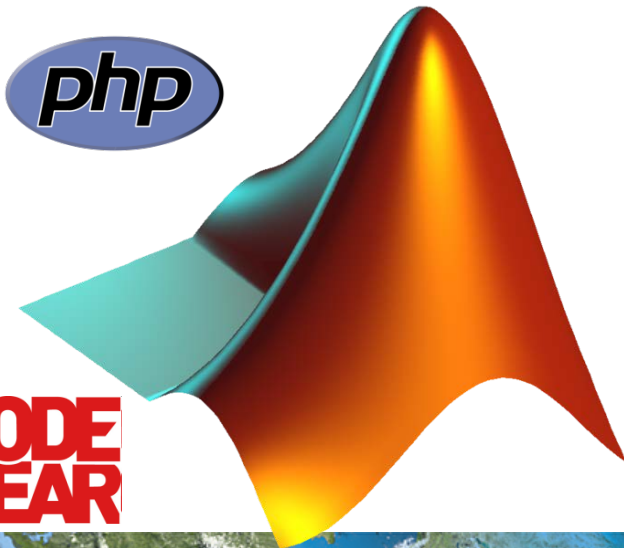
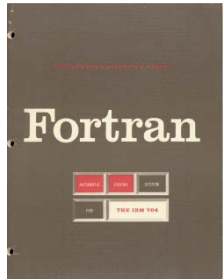
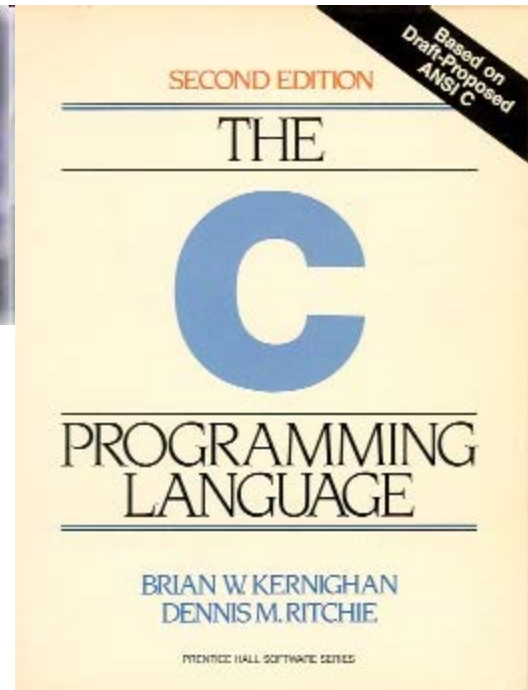
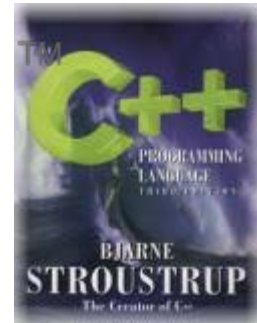
Charakteristiken

- Im DLR entwickelte Software ist in der Regel Nicht-Standard-Software (→ Individualsoftware)
 - Oft spezielle und spezifische Anforderungen
- Die „Entwickler“ haben oft keine Ausbildung in Software-Entwicklung
- Vielzahl an eingesetzten Software-Technologien
- Sowohl Open-Source als auch proprietäre Software
 - Viele Fragen und Probleme mit Open-Source-Lizenzen
- Große Anzahl an Software-Projekten
 - Überblick über existierende Software ist schwierig



Software im DLR





Software-Engineering-Strategie

Unterstützen der Software-Entwicklung im DLR

Methoden und Werkzeuge für die Software-Entwicklung

- Entwickeln von angepassten Prozessen für Wissenschaftler, inklusive Web-basierter Dokumentation
- Integration von Entwicklungswerkzeugen in die Arbeitsumgebungen der Wissenschaftler
- Werkzeuge sind *für jeden* verfügbar und einfach im Intranet zugreifbar
- Schulungen für verbreitete Entwicklungswerkzeuge und Software-Technologien



Software-Engineering-Strategie

Wissensmanagement

Austausch von Wissen und Informationen

- Netzwerk von Software-Engineering-Beauftragten
 - Informationsaustausch per Intranet und auf Workshops
- Software-Engineering-Wiki für Dokumentationen und Zusammenarbeit
- Software-Katalog
- Informationsbroschüre und Schulungen zu Open-Source-Lizenzrecht



Software-Engineering-Wiki

Startseite


DLR.Wiki


Bereiche ▾


Frage ans DLR

Erstellen

?





 Software Engineering

Seiten

Blog

BEREICHsverknüpfungen

Get Involved!

Topics

Tools

Best Practices

Software Project Manual

Events

SEITENHIERARCHIE


Get Involved!

Topics

Tools

Best Practices


Bereich konfigurieren ▾

Seiten 


Software Engineering

Erstellt von Pliewischkies, Andre, zuletzt geändert von Schlauch, Tobias am 24. Oktober 2014

Welcome to the *SoftwareEngineering.Wiki*!



The **SoftwareEngineering.Wiki** is the place to create, share and discuss software engineering content with colleagues on a working-level! We aim for an open and constructive exchange of ideas. Therefore, feel free to share your knowledge and encourage others to do so as well!

Please visit our Get Involved! section and  Subscribe to our RSS Feed before you start!

Please remember that the *SoftwareEngineering.Wiki* lives from your contributions!

Current Status:



At the moment this Wiki space is in its initial construction phase. We already established a basis structure and are currently in the process of adding more and more content. Please support us!

Search this Wiki Space

Contacts


- @Schlauch, Tobias
- @Haupt, Carina

Links

- Software-Engineering-Portalseite 
- Software Engineering Contacts 

Latest News

News-Einträge

 What wrong with writing

Get Involved!

Topics

Tools



Software-Engineering-Wiki

Programming-Guides für Entwickler

DLR.Wiki Bereiche ▾ Frage ans DLR Erstellen

Events

SEITENHIERARCHIE

- Get Involved!
- Topics
- Tools
- ▾ Best Practices
 - How-Tos
 - Experience Reports
- ▾ Programming Guides
 - Java Guide
 - ▾ **Python Guide**
 - New to Python?
 - Choosing the "right" Python Version
 - Python Development Tools
 - Python Style Guide
 - Code Example for Python
 - C++ Guide
- Software Project Manual

Bereich konfigurieren ▾ <<

Seiten / ... / Programming Guides

Bearbeiten Beobachten Teilen Extras ▾

Python Guide

Erstellt von Keutel, Christin, zuletzt geändert von Schlauch, Tobias am 13. Oktober 2014

This guide is not a Python beginner's manual. It aims to speed up the setup of a new Python project. It provides recommendations for essential development tools, a style guide for effective Python programming, and illustrates all recommendations using a full code example.

Please keep in mind that recommendations of this guide should not be considered as absolute rules. Particularly, the final decision should always be taken in context of your project requirements! If you have further questions, please contact one of the moderators of this guide.

Here you can retrieve a printable PDF version of this guide.

- New to Python?
 - Python for Beginners
 - Python for Developers
 - Python for Scientists
- Choosing the "right" Python Version
 - Relevant Python Interpreters and Versions
 - Current Recommendation
- Python Development Tools
 - Notes on Python Build Automation Support

Moderator(s)

- @Schlauch, Tobias
- @Meinel, Michael

Contributors

Keutel, Christin (vor 17 Tagen) ,
Schlauch, Tobias (vor 16 Tagen) ,
Lettich, Paul (vor 17 Tagen)

Complete list of available programming recommendations:

- Java Guide
- Python Guide
- C++ Guide



DLR Software-Katalog

<http://software.DLR.de>

- Verzeichnis für alle Software-Projekte im DLR
- Zugriff für DLR-Mitarbeiter und die Öffentlichkeit
- Für Open Source und proprietäre Software

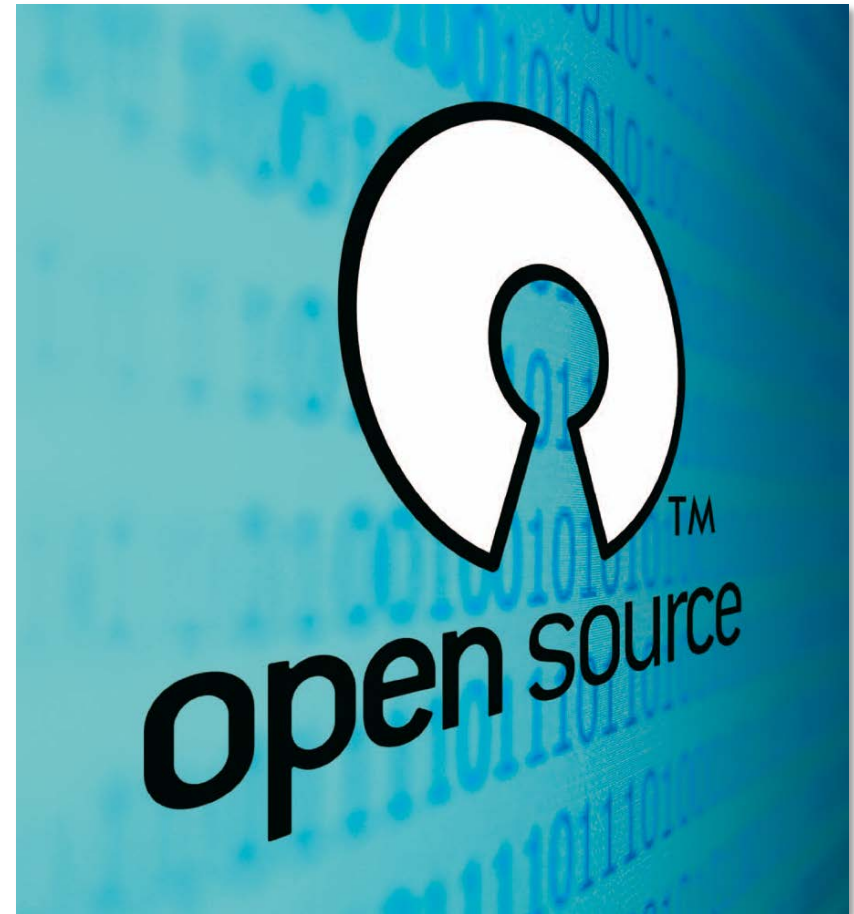


Open-Source-Strategie

DLR's Open-Source-Agenda

Sortiert nach Priorität

- Anleitungen und Hilfe für Open-Source-Lizenzen
- Kriterien für Auswahl von Open-Source-Software
- Best-Practices für eigene Open-Source-Projekte
- Migration nach Open Source



Open-Source-Strategie

Status im DLR

Bereits vorhanden

- Standard-Open-Source-Lizenzen ausgewählt
- Broschüre für Open-Source-Lizenzrecht
- Schulungen
- Hilfe und Unterstützung

Nächste Schritte

- „Standard“-Hosting von Open Source
 - Zur Zeit: SourceForge.net, Github, ~~Google Code~~, ...
- Prozesse für Auswahl und Freigabe
 - Teil des Qualitätsmanagementsystems



Open-Source-Strategie

Licenses

Im DLR werden viele Open-Source-Lizenzen verwendet

- AGPLv3, GPLv2, GPLv3, EPL, QPL, LGPLv2, LGPLv3, CDDL, MPLv2, Apache 2.0, BSD 2/3, MIT, Zlib, ZPLv2, Python 2.0, ...

Empfohlene Lizenzen (geprüft durch Rechtsabteilung)

- Simplified BSD License
- Apache License 2.0
- Eclipse Public License 1.0



DLR wird keine eigene Open-Source-Lizenz entwickeln

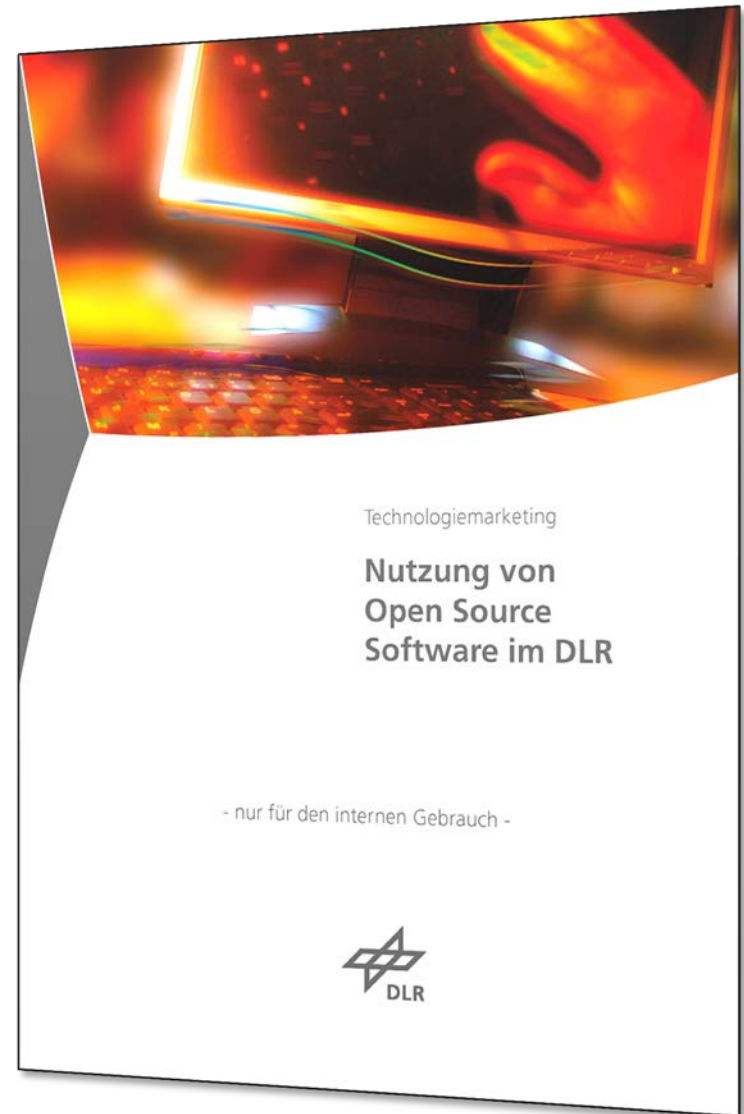
- Die OSI-zertifizierten Lizenzen reichen aus!



Open-Source-Broschüre

Lizenzrecht

- Grundlagenwissen über Open-Source-Lizenzrecht
- Entwickelt durch Anwaltskanzlei
- Finanziert und Herausgegeben von DLR-Technologiemarketing



Open-Source-Broschüre

Praktische Informationen

Checklisten


Infoboxen

Weitergabe veränderter Software

6

Lizenzen mit strengem Copyleft

GNU General Public License Version 2 (GPLv2)

-  ☐ Mitliefern des Lizenztexts (siehe Merkbox 1, S. 10)
- ☐ Zugänglichmachung des Quellcodes (s.o. Merkbox 2, S. 10)
- ☐ Vollständiger korrespondierender Quellcode (s.o. Merkbox 3, S. 11)
- ☐ Urhebervermerk (s.o. Merkbox 4, S. 11)


Beibehalten der vorbestehenden Urhebervermerke (s.o. Merkbox 4, S. 11)
Anbringen neuer Urhebervermerke

Merkbox 11

Bei der Veränderung der Software sind vorbestehende Urhebervermerke beizubehalten (s.o. Merkbox 4, S. 11) und neue Urhebervermerke in den hinzugefügten Dateien anzubringen.

Wie gestalte ich einen neuen Urhebervermerk?

Im Header der Quellcode-Dateien ist der Vermerk „© */Jahreszahl/*, Deutsches Zentrum für Luft- und Raumfahrt e.V., author: */Name/*“ anzubringen.

-  ☐ Disclaimer (s.o. Merkbox 5, S. 11)
- ☐ Änderungsvermerk

in den geänderten Quellcode-Dateien.

Merkbox 12

Wie gestalte ich den Änderungsvermerk?

Der vorbestehende Urhebervermerk ist beizubehalten und ein kurzer Hinweis auf die hinzugefügte/geänderte Funktion mit Datumsangabe und Namensnennung des Deutschen Zentrums für Luft- und Raumfahrt e.V. sowie des Autors der Änderung.

Der Urhebervermerk kann dabei wie in Merkbox 11, S. 24 aussehen.

-  ☐ Hinweispflichten bei interaktiven Kommandos

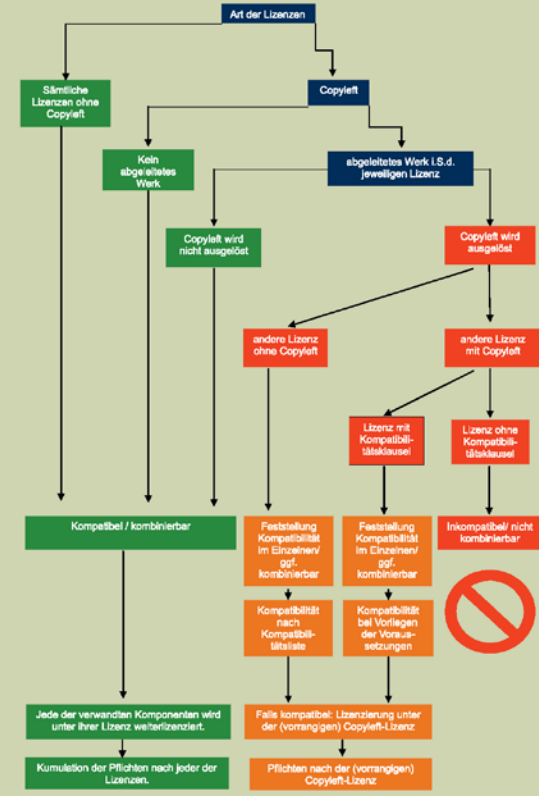
Open-Source-Broschüre

Entscheidungshilfen

Entscheidungs-
baum

11 Anhang

Übersicht: Wann sind die Fragen der Lizenzkompatibilität zu klären?



Open Source Schulungen

Lizenzrecht

- *„Rechtliche Aspekte der Open-Source-Nutzung im DLR“*
- Regelmäßige Standardschulung im DLR Bildungsprogramms
- Zusätzlich bei Bedarf für Institute oder Teams

Entwicklung

- *„Werkzeug-gestützte Software-Entwicklung“*
- Entwicklung mit Open-Source-Tools
- Standardschulung & nach Bedarf (*siehe oben*)



Open Source

Hilfe & Unterstützung

Hilfe und Unterstützung je nach Fragestellung

- Allgemeine Lizenzfragen, Schutzrechte
→ DLR-Technologiemarketing
- Rechtliche Unterstützung für Copyright- und sonstige Rechtsfragen
→ DLR-Rechtsabteilung
- Lizenzkompatibilität, Lizenzauswahl, Entwicklung
→ DLR-Simulations- und Softwaretechnik

Email

- opensource@dlr.de



Migration nach Open Source

Ersatz für kommerzielle Software

Langfristig: Ersatz / Alternativen für Standard-Software

- Microsoft Office
- Messenger und Telefonie (Microsoft Lync)
- Betriebssysteme
- ...

Kurzfristig: Alternativen für techn./wiss. Tools

- MATLAB
- Visualisierungs-Tools
- Compiler
- ...



Migration nach Python



Wissen für Morgen



Wichtige Anwendungssoftware

Wissenschaft und Technik

Viele kommerzielle Anwendungen sind noch Standard

- Microsoft Excel
- MATLAB
- IDL
- Fortran-Compiler

Der Weg nach Python...

- Einheitliche Sprache für viele Anwendungsgebiete
- Workshop „Migration nach Python“



Microsoft Excel

Verarbeitung tabellarischer Daten

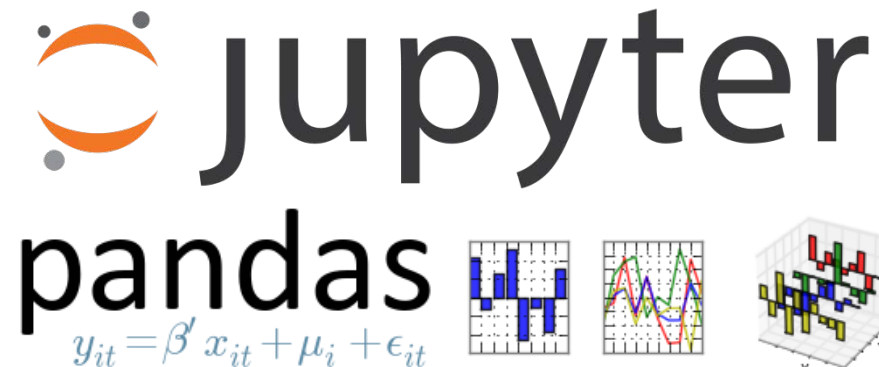
Wesentliche Funktionen

- Tabellen
- Sortier-, Gruppier-, Filterfunktionen
- Pivot-Tabellen
- Diagramme



Python-Alternative

- Jupyter / IPython
- pandas

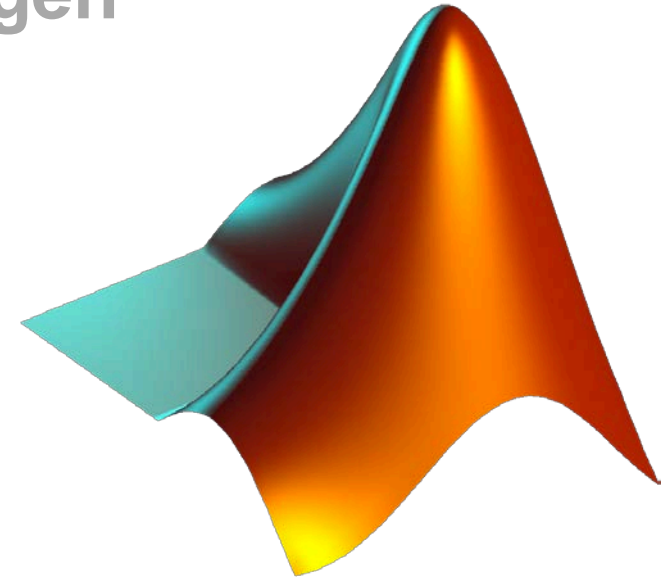


The MathWorks MATLAB

Numerische Matrixberechnungen

Wesentliche Funktionen

- Eigene proprietäre Programmiersprache
- Viele Anwendungs-Toolboxes z.B. Statistik, Signal- und Bildverarbeitung



Python-Alternative

- NumPy
- Matplotlib



Nützliche Quelle: wiki.scipy.org/NumPy_for_Matlab_Users



IDL – Interactive Data Language

Analyse und Visualisierung von Daten

Wesentliche Funktionen

- Array-basierte Programmiersprache
- Gute Bildverarbeitungsfunktionen



IDL

Python-Alternative

- IDL-nach-Python-Compiler PIKE

Torsion Analytics

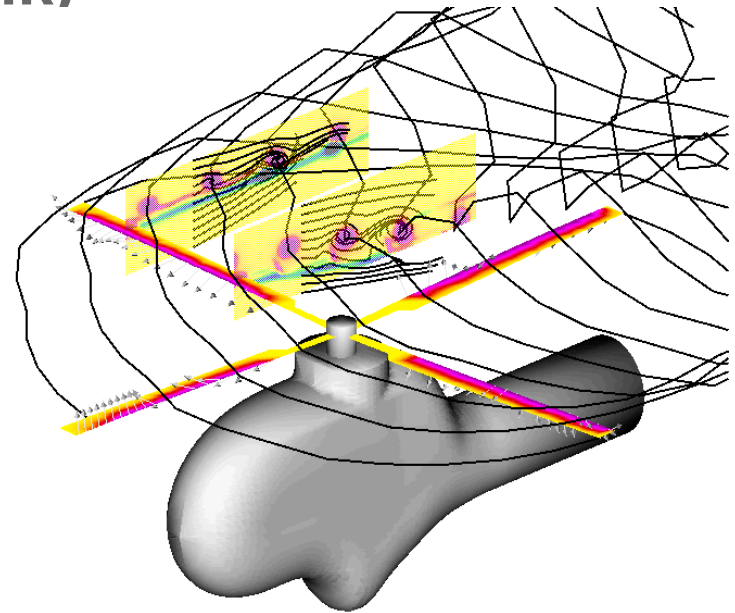


Python als Ersatz für Fortran?

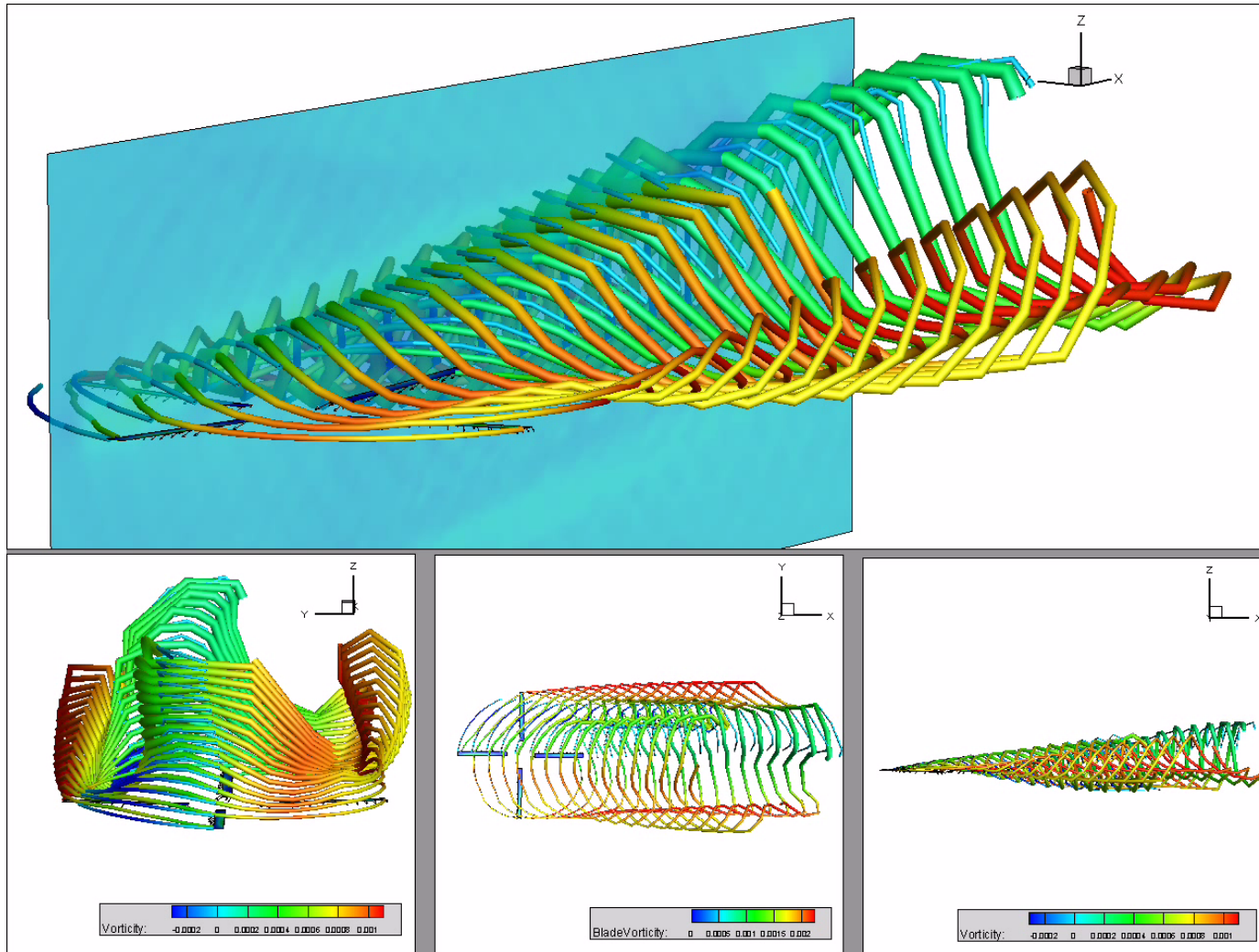
Beispiel: Simulation von Hubschrauber-Rotoren

Free-Wake (DLR-Flugsystemtechnik)

- Simulation dreidimensionaler Strömungen um einen aktiv gesteuerten Rotor eines Helikopters
- Code entwickelt 1994-1996
 - MPI-parallelisiert in Fortran
- Aufwendige Performance-Optimierung 2013-2014
 - MPI und Open ACC



Visualisierung der Wirbel



Kern-Schleifen von Free-Wake (Standard Python)

```
for iblades in range(numberOfBlades):
    for iradial in range(1, dimensionInRadialDirection):
        for iazimutal in range(dimensionInAzimualDirectionTotal):
            for i1 in range(len(vx[0])):
                for i2 in range(len(vx[0][0])):
                    for i3 in range(len(vx[0][0][0])):
                        # wilin-Aufruf 1
for iblades in range(numberOfBlades):
    for iradial in range(dimensionInRadialDirection):
        for iazimutal in range(1, dimensionInAzimualDirectionTotal):
            for i1 in range(len(vx[0])):
                for i2 in range(len(vx[0][0])):
                    for i3 in range(len(vx[0][0][0])):
                        # wilin-Aufruf 2
for iDir in range(3):
    for i in range(numberOfBlades):
        for j in range(dimensionInRadialDirection):
            for k in range(dimensionInAzimualDirectionTotal):
                x[iDir][i][j][k] = x[iDir][i][j][k] +
                                     dt * vx[iDir][i][j][k]
```



Free-Wake

Performance-Vergleich Fortran – Python

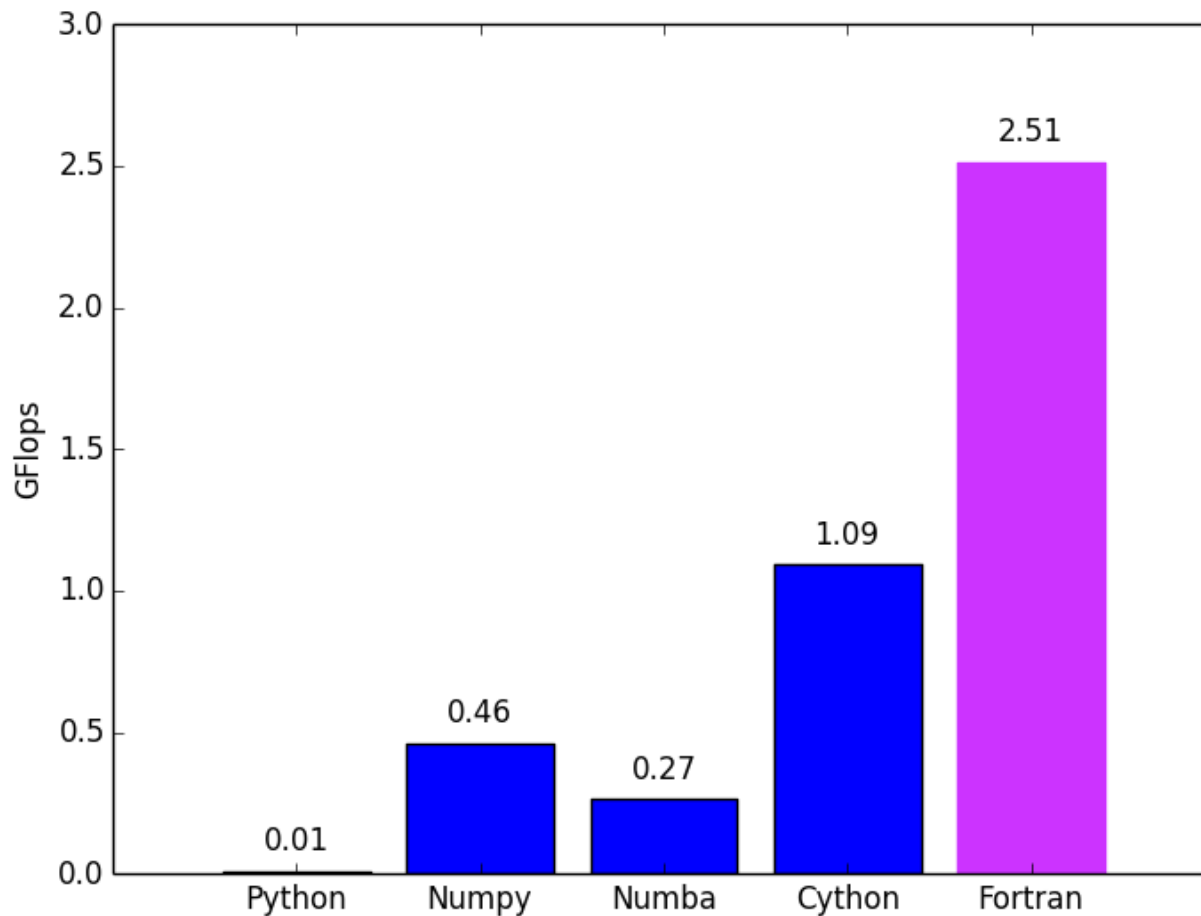
Vergleich der hoch-optimierten Fortran-Version mit parallelen Python-Versionen

- Multi-core CPUs
 - Cython mit OpenMP
 - Python-Bindings für Global Array Toolkit
- GPGPUs
 - NumbaPro



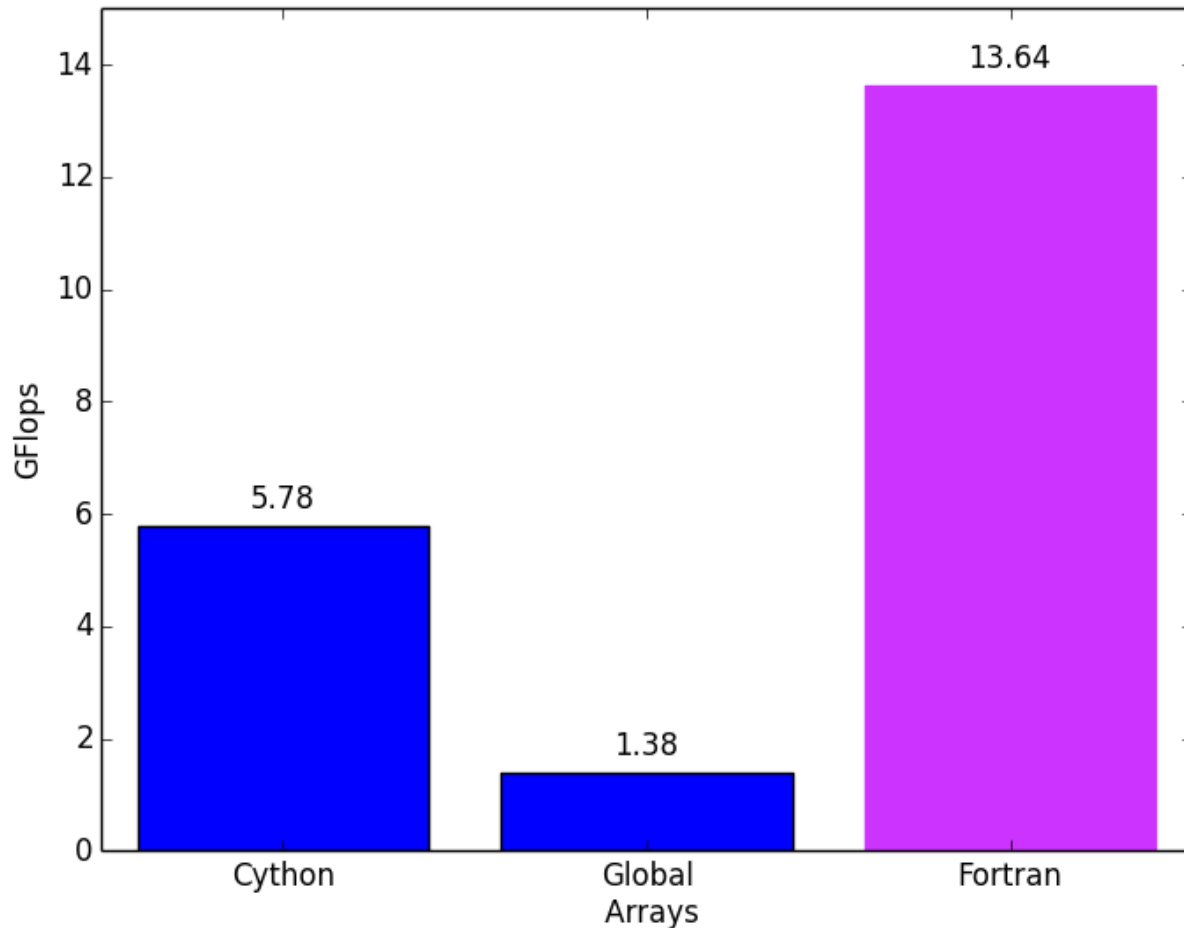
Performance-Tests

Single-Core Performance (Xeon E5645, 6 Cores)



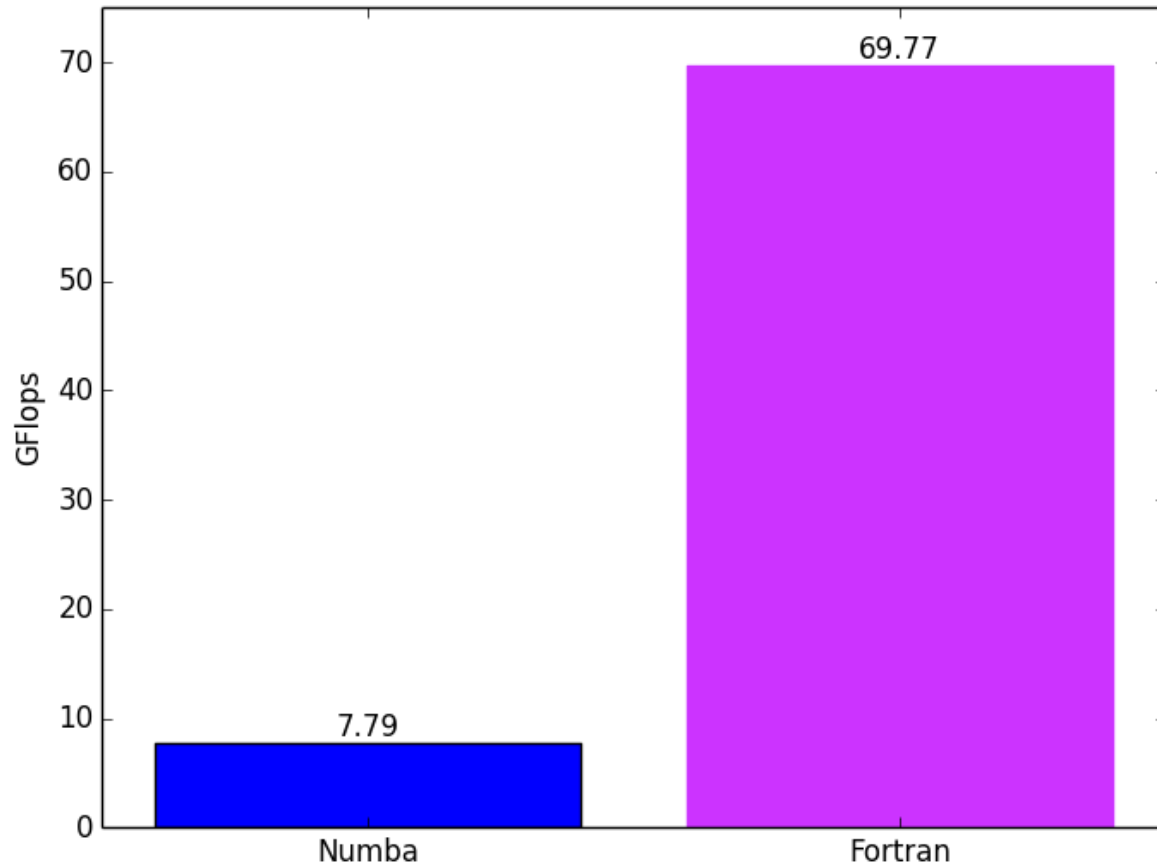
Performance-Tests

Multi-Core Performance (Xeon E5645, 6 Cores)



Performance-Tests

GPGPU Perf. (NVIDIA Tesla C2075, 448 CUDA-Cores)



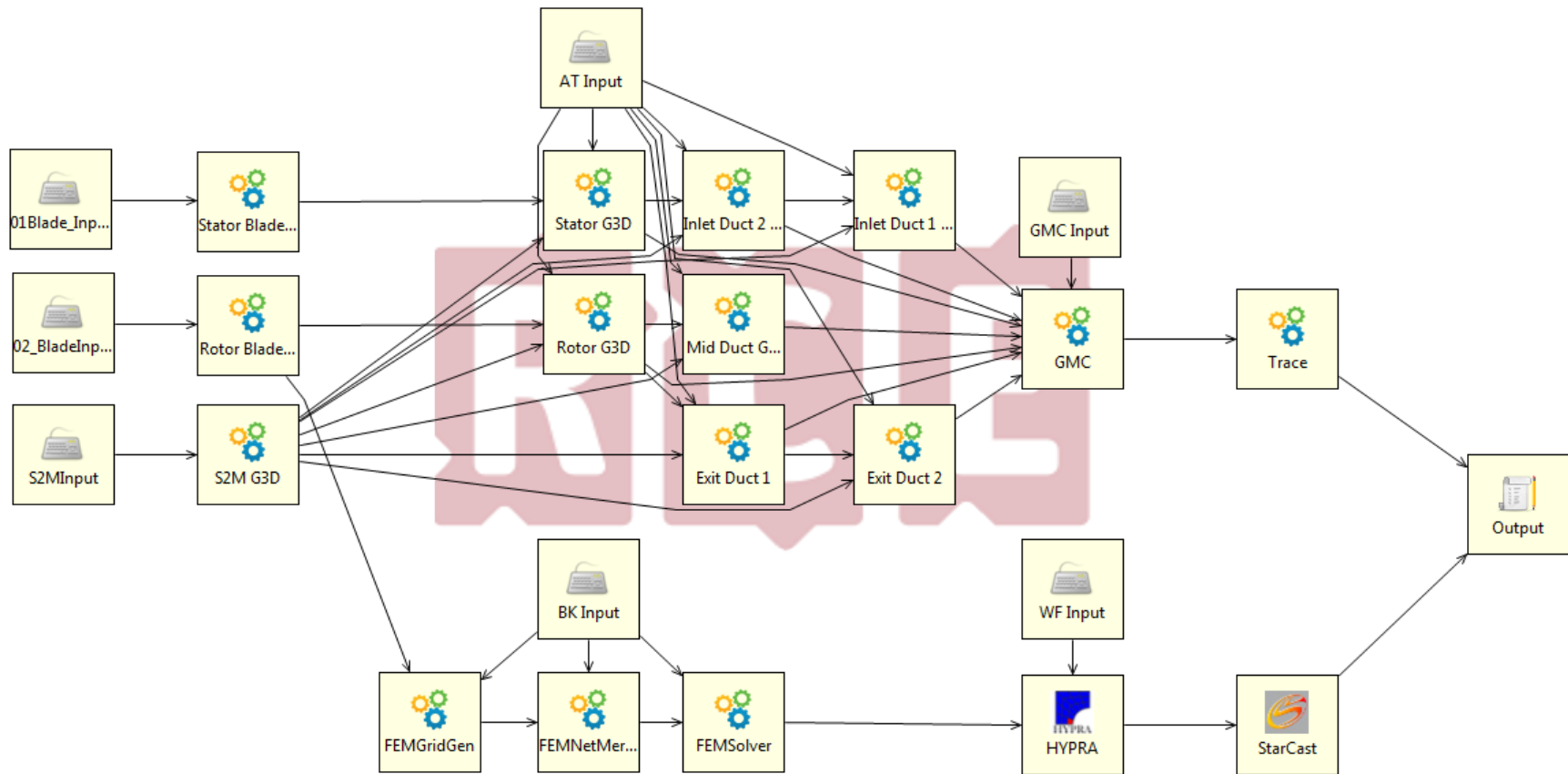
Beispiele für Open-Source-Projekte des DLR



Wissen für Morgen



Beispiel Software für multidisziplinären Entwurf Remote Component Environment (RCE)



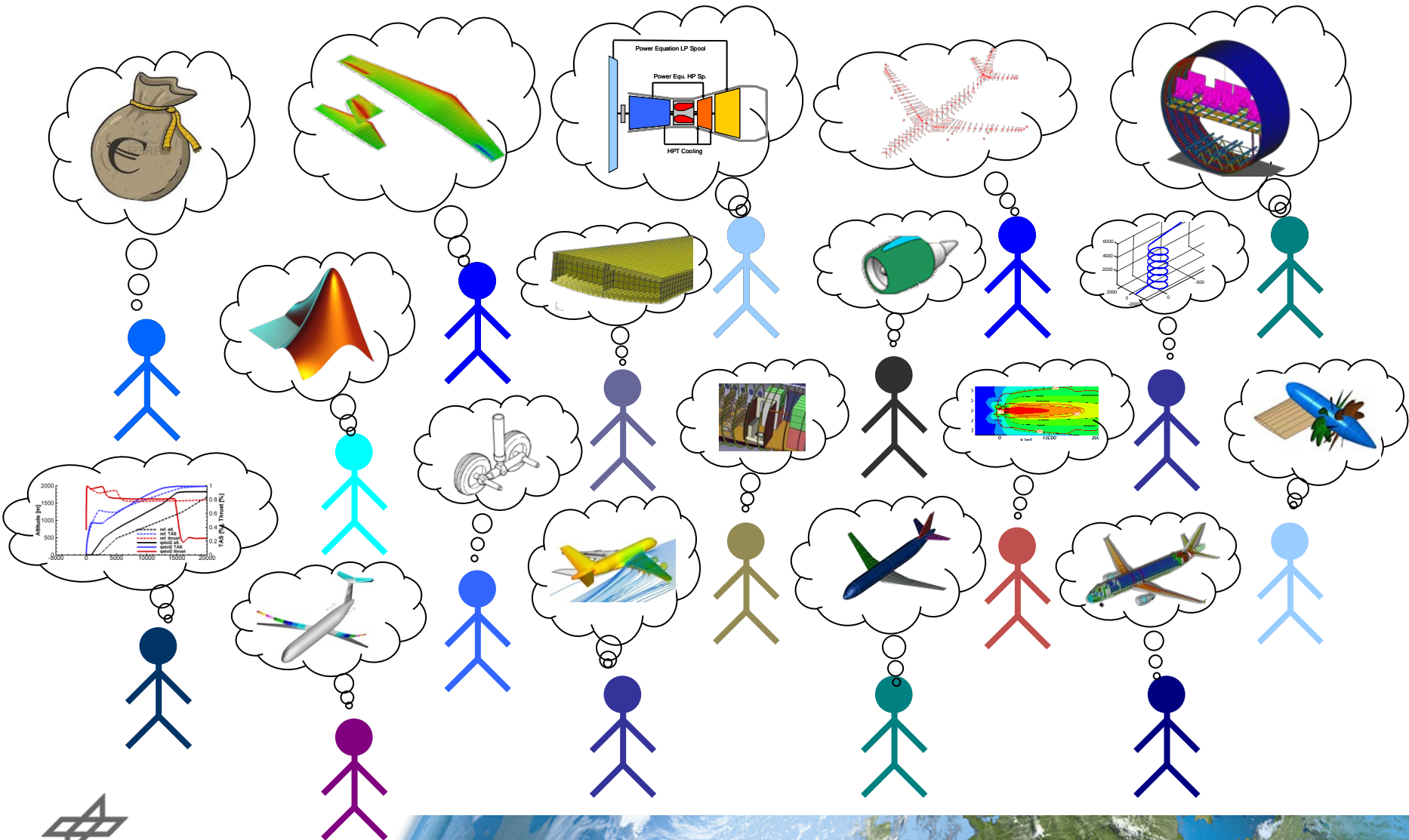
Entwurf in der Luftfahrt

Ziel: Neue Flugzeugkonfigurationen

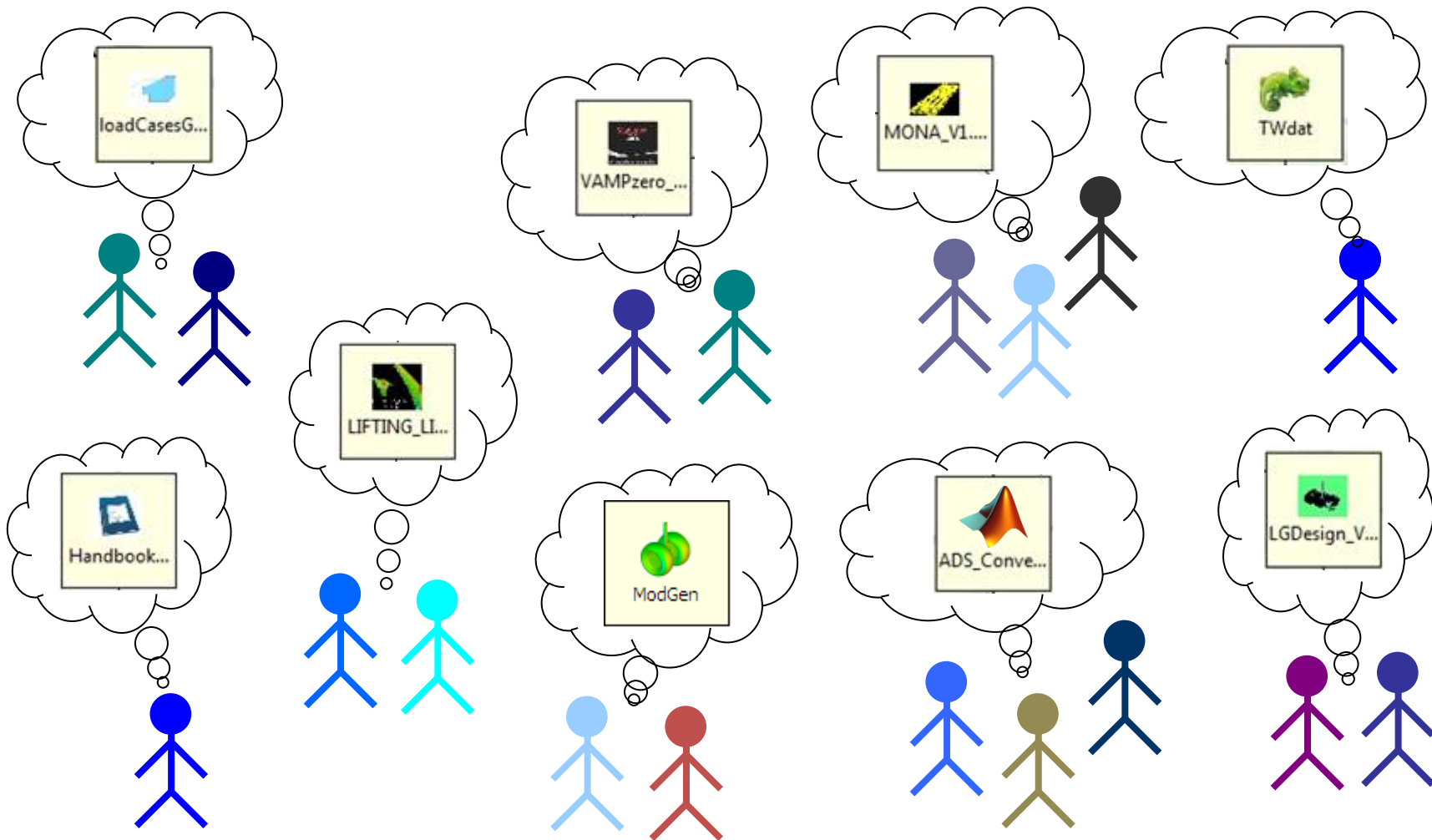
- geringere Betriebskosten
- umweltverträglicher



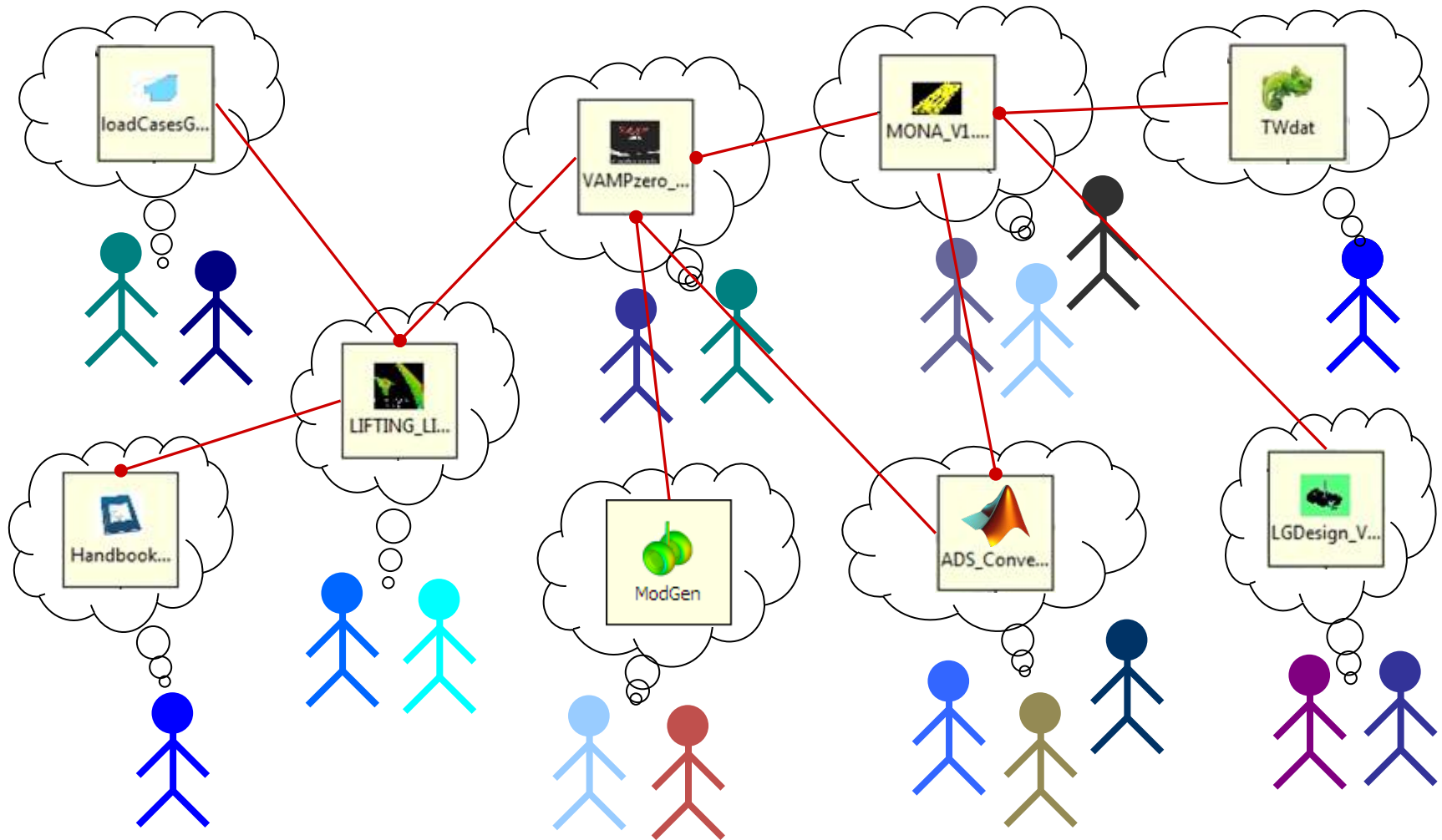
Multidisziplinärer Entwurf



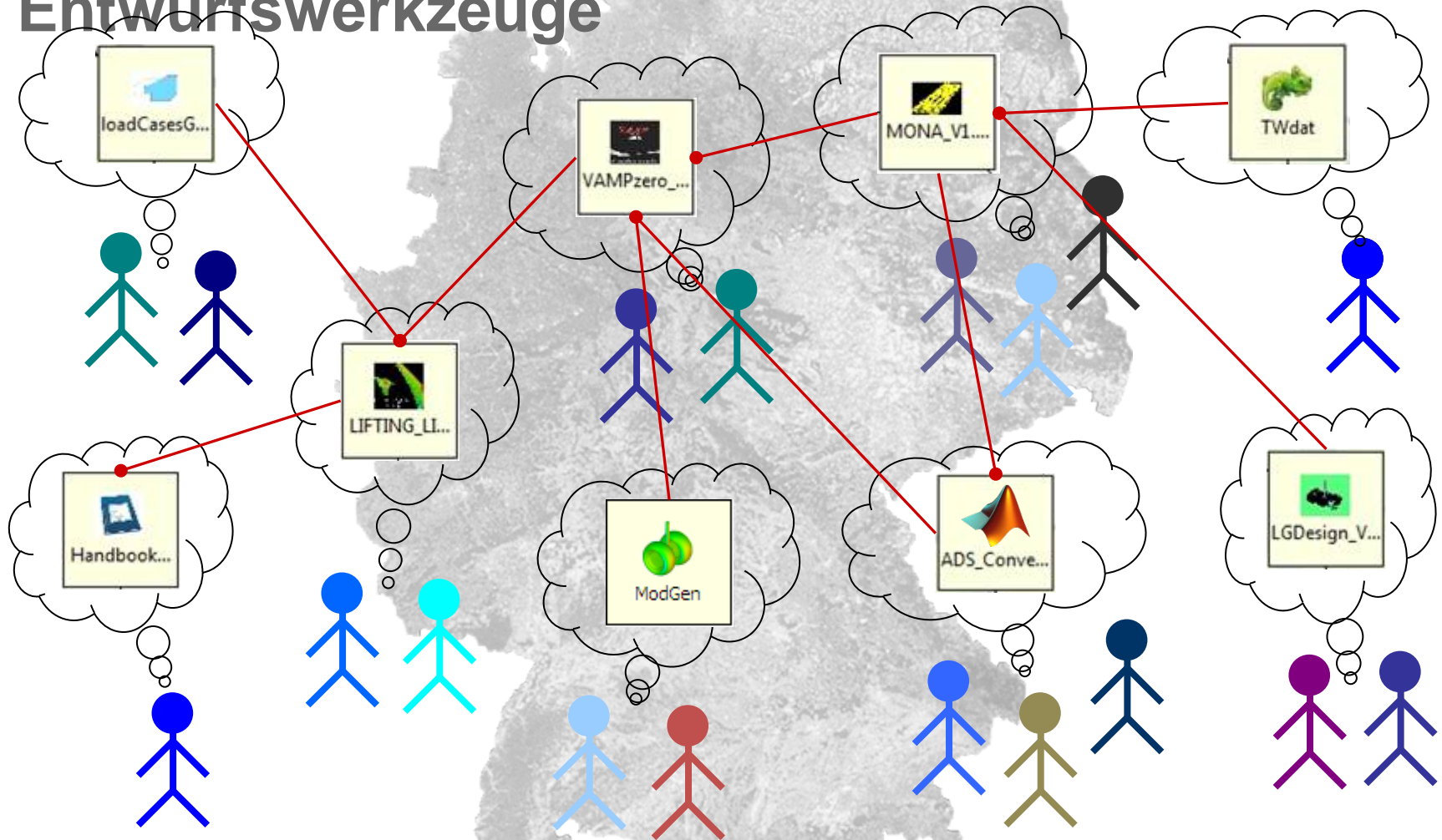
Multidisziplinäre Entwurfswerkzeuge



Kopplung multidisziplinärer Entwurfswerkzeuge



Kopplung verteilter, multidisziplinärer Entwurfswerkzeuge



Integrationsumgebung RCE

Softwaresystem

- Zur Modellierung von Entwurfsprozessen
- Zur automatisierten Ausführung von verteilten, verketteten Entwurfswerkzeugen



Historie

- Entwickelt für den Schiffbau im Drittmittelprojekt SESIS
- Im und fürs DLR weiterentwickelt und inzwischen in Luftfahrt, Raumfahrt und Verkehr eingesetzt
- Externe Nutzer



Integrationsumgebung RCE

Open Source



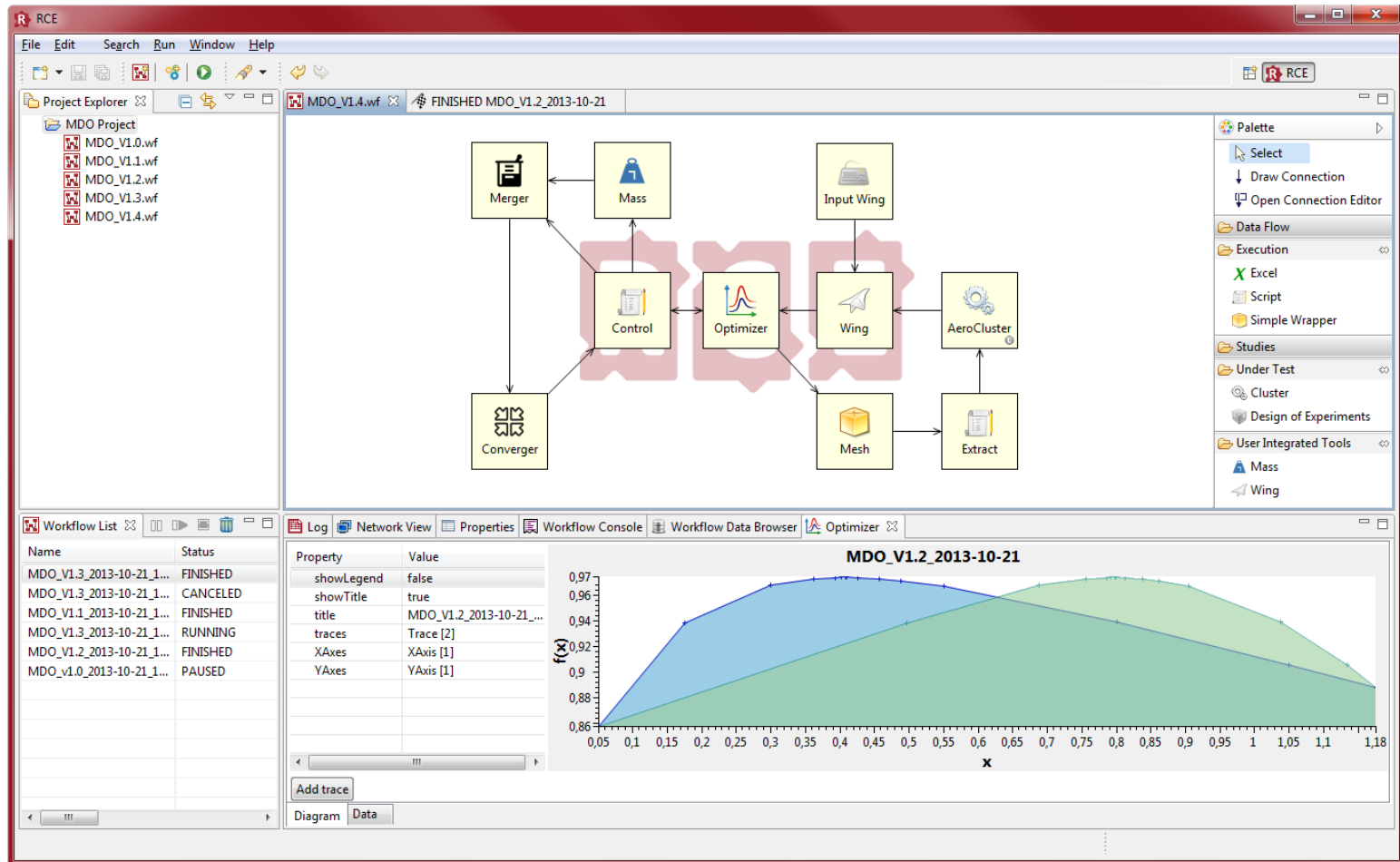
Open Source unter der Eclipse Public License (EPL)

Links

- Github: <https://github.com/rcenvironment>
- Website: <http://rcenvironment.de>
- Twitter: @rcenvironment
- Youtube: <http://youtube.com/rcenvironment>

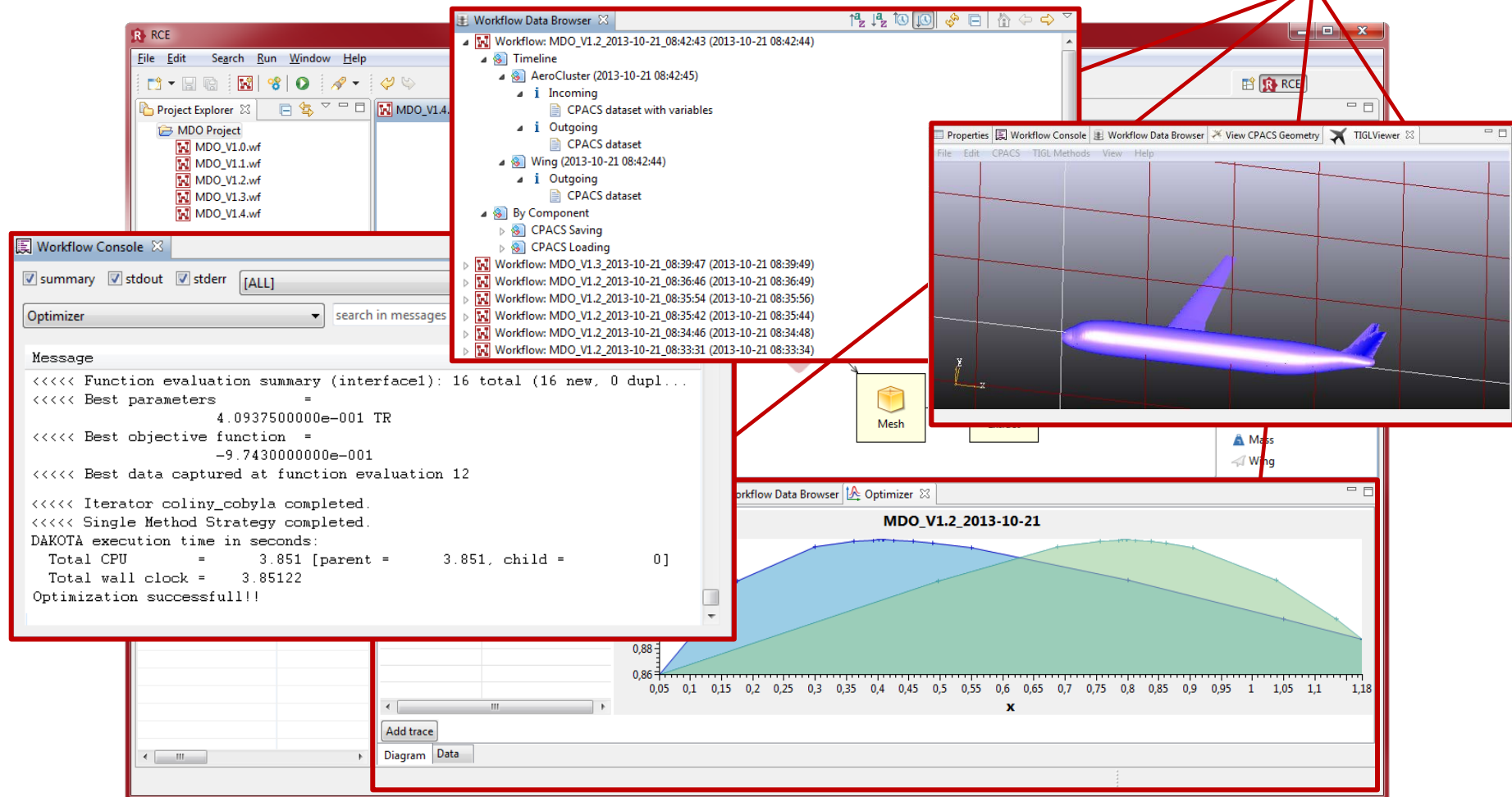


Grafische Benutzeroberfläche von RCE

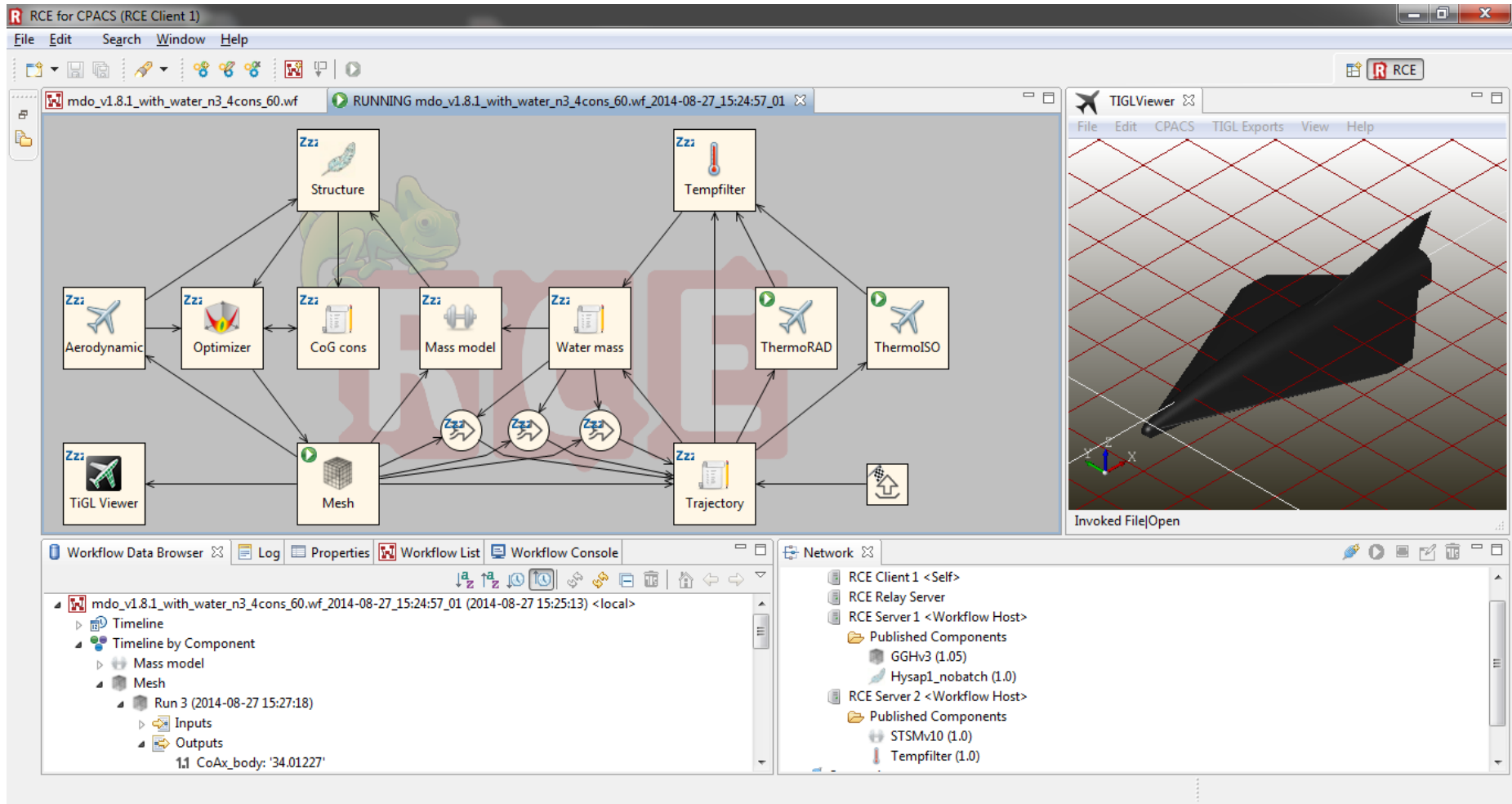


Grafische Benutzeroberfläche

Ergebnisdarstellung
während und nach einer
verteilten Ausführung der
Werkzeuge



RCE in der Raumfahrt

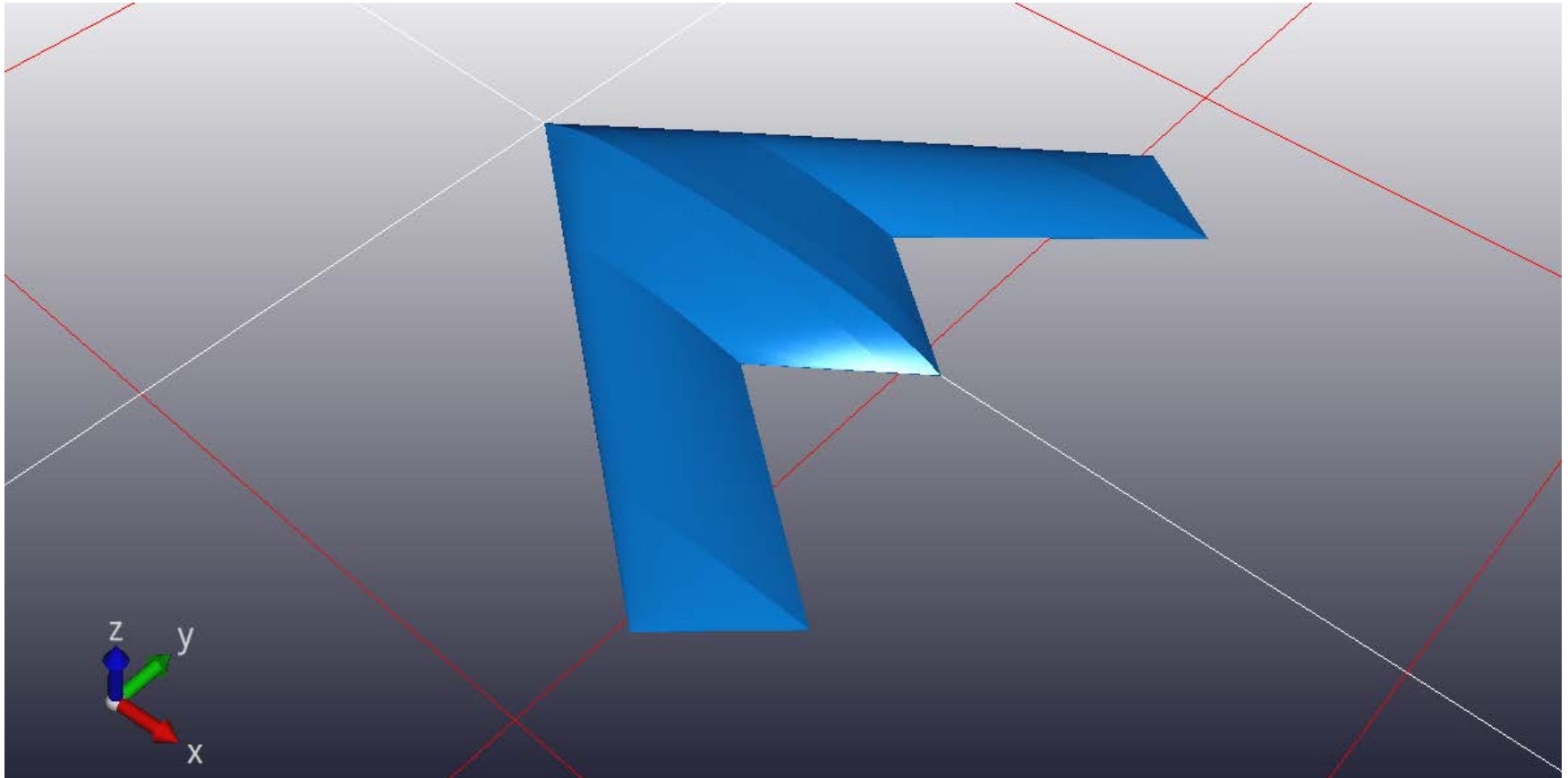


RCE in der Luftfahrt



Beispiel Numerische Bibliotheken

Geometriebibliothek TiGL



Geometriebibliothek TiGL

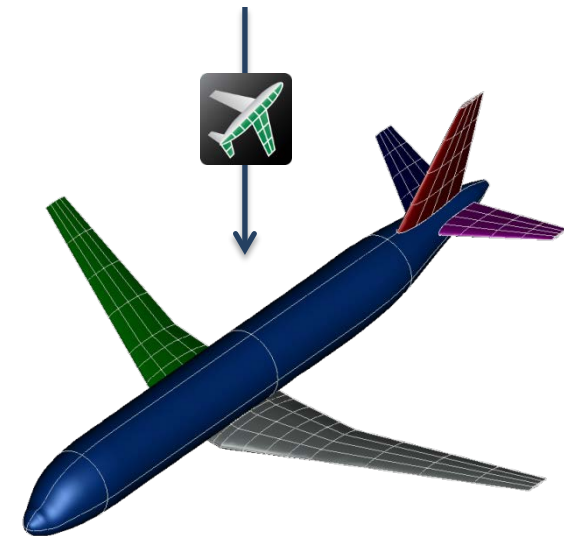
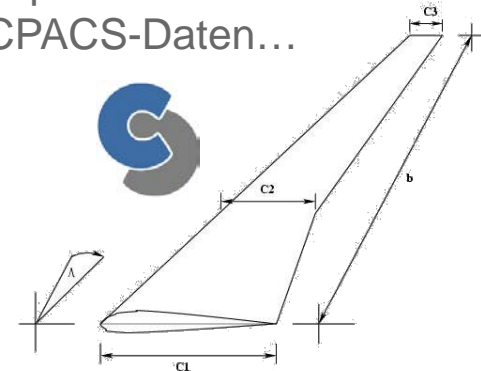
Herausforderungen

Berechnen einer 3D-Geometrie
aus parametrisierten Daten (DLR-
Format *CPACS*)

Ausschnitte extrahieren

- Punkte auf der Oberfläche
- Punkte auf das Flugzeug projizieren
- Schnitte durch den Rumpf
- Schnitte mit Linien
- Schnittkurve zwischen Flügel und Rumpf
- ...

Von parametrisierten
CPACS-Daten...



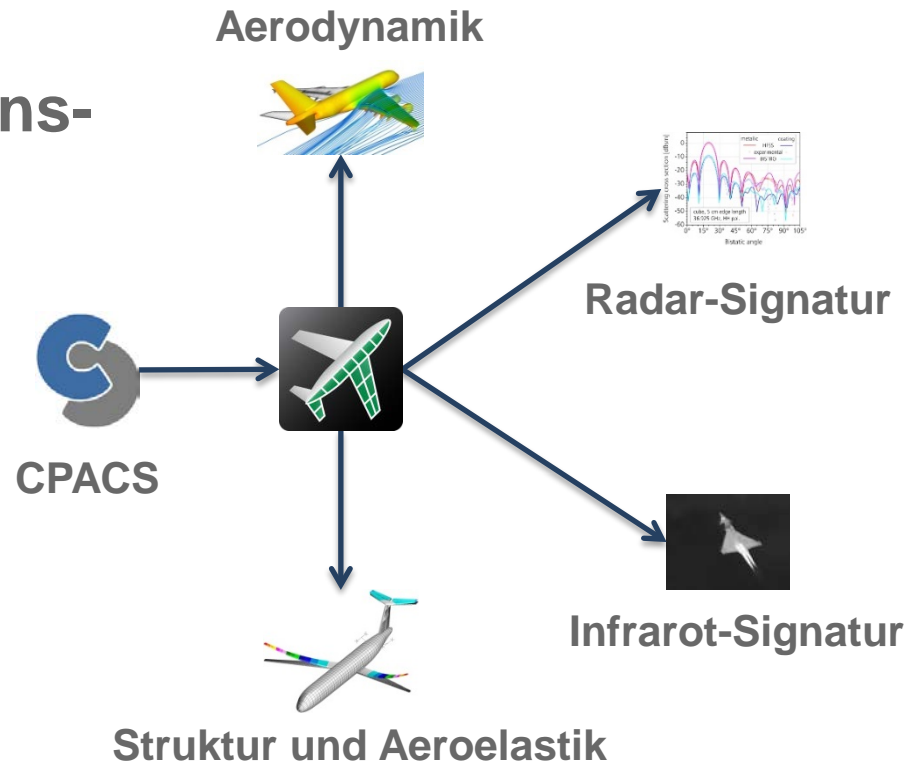
...zur 3D-CAD-Geometrie

Geometriebibliothek TiGL

Geometrieoperationen für Flugzeugentwurf

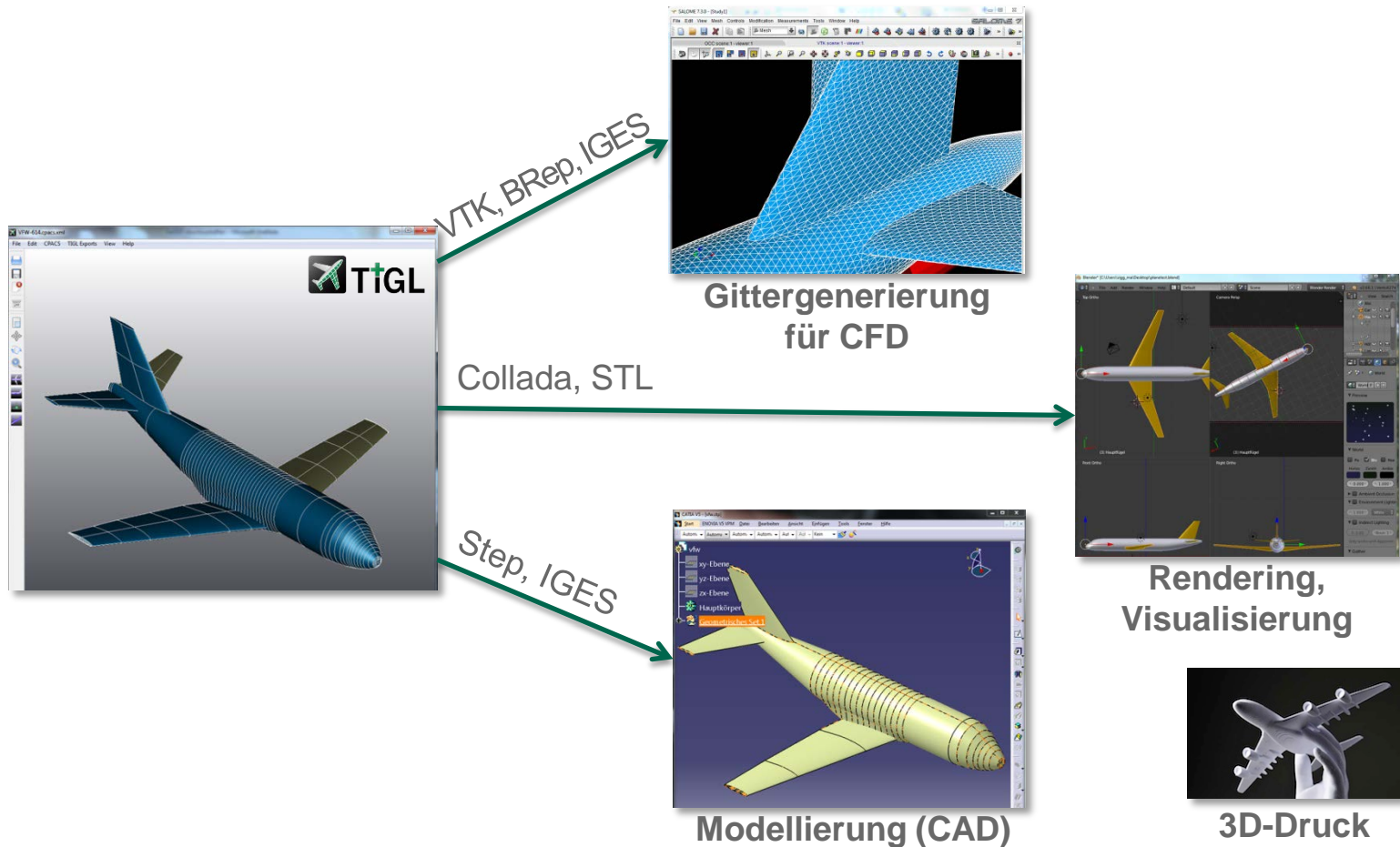
Benutzt für viele Simulationscodes des DLR

- *Multi-purpose*
- *Multi-language*
- *Multi-platform*



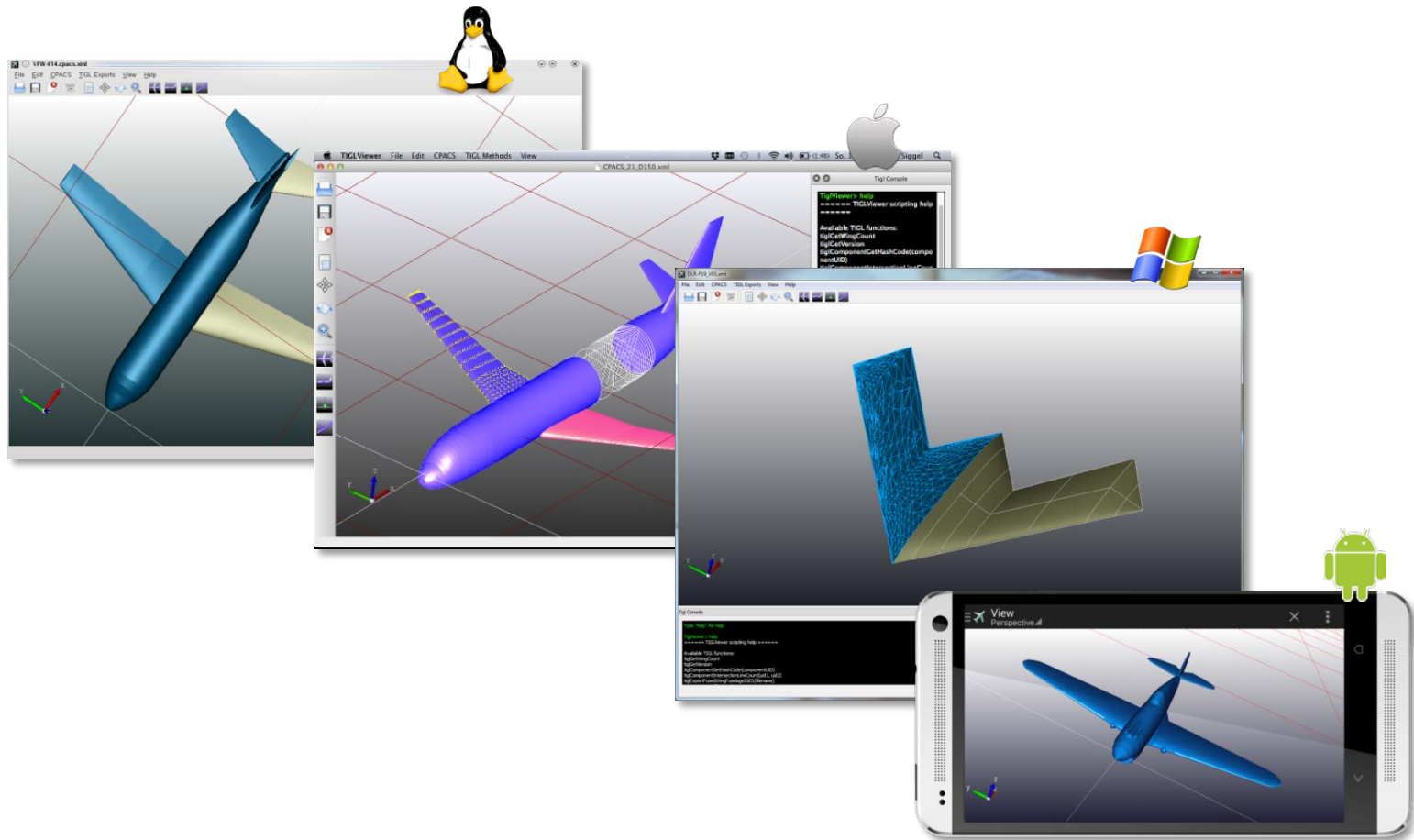
Geometriebibliothek TiGL

Export-Funktionen



Geometriebibliothek TiGL

Plattformunabhängige Betrachter



Geometriebibliothek TiGL

Open Source



Open Source unter der Apache License 2.0

Links

- Github: <https://github.com/DLR-SC/tigl>
- Android-App:
<https://play.google.com/store/apps/details?id=de.dlr.sc.tiglviewer.android>



Vielen Dank!

Fragen?

Andreas.Schreiber@dlr.de
www.DLR.de/sc | [@onyame](#)

